Chapter 1. PACS Spectroscopy Launch Pad I

1.1. Introduction

Welcome to the PACS data reduction guide (*PDRG*). We hope you have gotten some good data from PACS and want to get stuck in to working with them. The PDRG explains the running of the data reduction pipelines for PACS spectroscopy in HIPE, and how to interact with the data. This first chapter—the *PACS Spectroscopy Launch Pad I*—is a "ReadMeFirst" to working with PACS spectroscopy.

1. How do I get and save a PACS spectroscopy observation?

2. How do I understand what type of observation I have?

3. What are the science products in my observation?

4. How can I plot my spectra and view my cubes?

5. What scripts or cookbooks can help me work with PACS spectroscopy?

The second chapter—Chapter 2, the *PACS Spectroscopy Launch Pad II*—is about the pipeline, and should be read by everyone working with PACS spectroscopy data (even if they do not want to run one of the pipelines in HIPE themselves).

1. What are the pipelines available for reducing PACS data

2. Do I *need* to re-run the pipeline (and if so, which one?)

3. Is it anyway *useful* to re-run the pipeline to improve some of the results?

4. Which are the crucial pipeline tasks?

5. I have a point source: what do I do next?

6. I have an extended source: what do I do next?

7. Where do I go to learn about the errors in my spectra?

The rest of the PDRG is:

- Chapter 3: gives more detail on the calibration files used in the pipeline processing, on getting and saving data, on where the pipeline scripts are, and an explanation of the differences between these interactive pipeline scripts. *An observation gotten from the HSA will only have been processed through one type of pipeline: this chapter is also useful for understanding whether you should consider running one of the other pipelines for your observation.*

- Chapter 4: concerns the beginning (Level 0—0.5) and the end (post-processing) parts of the reduction of PACS data, which are almost the same for all the pipelines. Chapter 5: explains the data reduction from Level 0.5 to 2 for chopNod mode observations. Chapter 6: explains the data reduction from Level 0.5 to 2/2.5 for unchopped mode observations.

- Chapter 7: explains some of the more crucial pipeline tasks: wavelength regridding; flatfielding; using the different cube mosaicking tasks (drizzle, interpolate, project); transient correction for the unchopped mode; background subtraction for the unchopped mode; and the unique tasks of the pointing offset pipeline. The errors in the final PACS spectroscopy products are also explained here.

Whether you run the pipeline or not, read this chapter to understand the effect these crucial tasks can have on the appearance of the spectra in the science products of the pipeline.

- Chapter 8: explains the post-pipeline tasks that are provided for point sources and semi-extended sources, and which can be run after any pipeline script.
- Chapter 9: explains the post-processing tasks to deal with mapping observations, creating mosaic cubes of various type, the differences between the various types of cubes, and what each one is best used for. *Whether you run the pipeline or not, read this chapter to understand why the different science-level cubes look the way they do.*
- Chapter 10: includes scripts and general information about plotting PACS data for diagnostic reasons during the pipeline processing, describes the viewers for inspecting PACS data (spectrally and spatially); and explains how to work at a deeper level with some of the datasets of the cubes.

Additional reading can be found on the HIPE help page, which you can access from the HIPE *Help#Help Contents* menu. This covers the topics of: HIPE itself, I/O, scripting in HIPE, and using the various data inspection and analysis tools provided in HIPE. You can also consult the PACS calibration pages on the Herschel Science Centre site, where the Observer's Manual and calibration documentation and information are provided (herschel.esac.esa.int/twiki/bin/view/Public/PacsCalibrationWeb). Details of the spatial and spectral calibrations applied to PACS data is **not** covered in the PDRG.

Text written *like this* refers to the class of a product (or to any product of that class). Different classes have different (java) methods that can be applied to them and different tasks will run (or not) on them. See the **Scripting Guide**, to learn more about classes. Text written `like this` refers to the parameters of a task.

1.1.1. Terminology

The following definitions will be useful to know:

- **HIPE** Herschel Interactive Processing Environment
- **DAG**: the HIPE Data Analysis Guide (this explains the general HIPE data analysis tools)
- **SG**, the Scripting Guide (a guide to scripting in HIPE)
- **PACS URM** the User's Reference Manual, describes the PACS tasks, their parameters and their function
- **PPE** the PACS Products Explained, which is about the products you get from the HSA or which you produce while pipeline processing PACS data
- **HSA, HSC** Herschel Science Archive and Herschel Science Centre
- **AOT** different PACS observing modes were programed with different Astronomical Observing Templates by the proposer
- **Level 0** products are raw and come straight from the satellite
- **Level 0.5** products have been partially reduced, spatially calibrated, and corrected for instrument effects by tasks for which no interaction is required by the user
- **Level 1** products have been more fully reduced, some pipeline tasks requiring inspection and maybe interaction on the part of the user
- **Level 2** products are fully reduced and flux calibrated, including tasks that require the highest level of inspection and interaction on the part of the user. Post-pipeline processing for point and extended sources is done on these data

- **Level 2.5** products can be found for unchopped range observations only. For this AOT, the on-source and off-source observations are taken separately: therefore each is reduced to Level 2, and then the off-source data are subtracted from the on-source data and the resulting products placed in Level 2.5 in the on-source observation. The post-pipeline processing can be done on these data

- **Level 3**: can be found for chopNod, pointed, full SED-coverage observations only. These are spectrum tables, which are a concatenation of data from the two or three observations that were taken to cover the entire PACS spectral range for a target

- **SPG**: standard product generation: the data reduced with a standard pipeline by the HSC. For each major version of HIPE a new SPG is run (this taking a few months of time). The SPG version of any observation is shown in the HSA results tab, in the "Summary" tab of the Observation Viewer, and also in the Meta datum "creator" of the *ObservationContext*. "SPG 13" is the SPG using the pipeline scripts of HIPE 13.

- **spaxel**: a spatial pixel, the spatial unit of an Integral Field Unit (IFU). Each spaxel contains the spectrum from one unique "pixel" on the sky. The native spaxel size of PACS is 9.4x9.4 arcsec; when you mosaic together cubes the resulting cube's spaxels are smaller, but they are still called "spaxels".

1.2. Getting and saving PACS observations

Herschel data are stored in the **HSA**.

- They are identified with a unique number known as the Observation ID (**obsid**). You can find the obsid via the HSA.

- They can be downloaded directly into HIPE, or one at a time to disc, or many as a tarball.

- The data you get from the HSA is an **Observation Context**, which is a container for all the science data and all the auxiliary and calibration data that are associated with an observation, and includes the **SPG** products. The entire observations is stored on disk as individual FITS files organised in a layered directory structure. The *ObservationContext* you load into HIPE contains links to all these files, and GUIs are provided to navigate through the layers.

There are several ways to **get and save observations from the HSA or disk** via HIPE. It does not matter which method you use.

- **Get the data directly from the HSA into HIPE on the command line, and then save to disk**:

```
obsid = 134..... # enter your own obsid
# To load into HIPE:
myobs = getObservation(obsid, useHsa=True)

# To load into HIPE and at the same time to save to disk
# A: to save to the "MyHsa" directory (HOME/.hcss/MyHsa)
myobs = getObservation(obsid, useHsa=True, save=True)
# B: to save to your "local store" (usually HOME/.hcss/lstore)
myobs = getObservation(obsid, useHsa=True)
saveObservation(myobs)
# C: to save to another disk location entirely, use:
pooll = "/Volumes/BigDisk/"
pooln = "NGC3333"
myobs = getObservation(obsid, useHsa=True)
saveObservation(myobs, poolLocation=pooll, poolName=pooln)
```

  See the *DAG* sec. 1.4.5 for more information on getObservation (for example, how to log on to the HSA before you can get the data, and more about "MyHSA"), and Section 3.2. For full parameters of getObservation, see its URM entry.

- **To get the data back from disk into HIPE:**

A and B: If you saved the data to disk with the default name and location (either [HOME]/.hcss/MyHSA or [HOME]/.hcss/lstore) then you need only specify the obsid:

```
obsid = 134...... # enter your obsid here
myobs=getObservation(obsid)
```

C: If you used saveObservation with a `poolName` and/or `poolLocation` specified:

```
obsid = 134...... # enter your obsid here
pooll = "/Volumes/BigDisk/"
pooln = "NGC3333"
myobs=getObservation(obsid, poolLocation=pooll, poolName=pooln)
```

In Chapter 3 you can find a longer summary of getting and saving. The PACS pipelines use these command-line methods. To learn about the GUI methods, see chap. 1 of the DAG.

1.3. What type of observation do I have?

PACS spectrometer observations were executed following a set of observing templates: the **AOTs**. Different AOTs may need to be be reduced with different pipelines.

- **chopNod, unchopped, or wavelength switching AOTs**: the difference between these lies in the observing technique used to sample the telescope+astronomical background. chopNod was the most common AOT. Unchopped was used for sources in crowded fields where chopping-nodding was not possible; for unchopped rangeScans the observers had to specify a separate observation for the off-source (background) position, for unchopped lines scans the on- and off-source were taken within one observation. Wavelength switching was also for crowded fields but this mode was discontinued a few months into the mission.

- **Line or Range spectroscopy**: with a wavelength ranges that encompasses one unresolved line only (Line); an observer-defined wavelength range (Range); or the full spectral range of PACS (SED, which is part of Range).

- **Pointed, undersampled mapping (tiling), Nyquist mapping, or oversampled mapping**: refers to the pointing mode: a single pointing; a large-step raster to cover a large field-of-view (tiling); a smaller-step raster to achieve a Nyquist spatial sampling of the beam; or very a fine-sampling raster to oversample the PSF.

  For the three mapping modes, whether undersampled or not dictates which pipeline task can be used to combine the rasters into a single, mosaic cube.

A more detailed summary for each mode is provided in Section 3.3. Note that all observations always contain data from the blue and the red camera.

Once you have downloaded an observation into HIPE you can find all relevant AOT information with:

```
obsSummary(myobs)
```

or look at the **pacsObsSummary** in the ObservationContext (use the Observation viewer on your "obs"). In the summary text look for the section "AOT and instrument configuration", e.g.

```
AOT and instrument configuration:
    AOT:               PacsLineSpec
    Mode:              Mapping, Chop/Nod
    Bands:             B3A R1 (prime diffraction orders selected)
    Is bright:         YES (shortened range mode)
    Raster lines:      5
    Raster columns:    5
    Raster line step:  14.5 (arcseconds)
```

```
Raster point step: 16.0 (arcseconds)
Chopper:           large throw
Nod cycles:        1
```

and there you will find:

- *AOT* (line or range)
- *Mode* (mapping or pointed; and unchopped, wavelength switching, or chopNod)

  For Mapping modes, note down the size of the steps (Raster line or point step) and number of steps (Raster lines or columns); this information will be useful later.

1.4. What are the science products in my observation?

1.4.1. The quality report

The quality report comes in the form of a **quality** and/or **qualitySummary** in the ObservationContext. They both contain the same information, but the qualitySummary (if present) also contains a report created after a manual check of the observation has been done at the HSC.

Click on the +quality/+qualitySummary from withim the Data tab of the Observation viewer and the report viewer will open to the right of that tab. The most important things to check are the Quality flags and comments: noting that by SPG 14 most observations have no flags and if there are no comments to make, the comments part will be blank. See also the quality documentation on herschel.esac.esa.int/twiki/bin/view/Public/DpKnownIssues.

1.4.2. The science-quality cubes

The cubes produced at the end of the SPG pipeline are found in **Level 2** for most observations, **Level 2.5** for the on-source observation of an unchopped range pair. Since there can be, within any observation, several pointings and multiple wavelength ranges, all the related cubes are gathered together in *context*s: one *context* for the red camera and one *context* for the blue camera. In addition, PACS produces different types of cube depending on the pointing mode adopted, and there are always separate sets of red and blue *context*s for each type of cube.

- Double-click on your observation in the *Variables* panel (or right-click and select the **Observation Viewer**). The viewer will open in the *Editor* pane of HIPE.
- In the directory-like listing on the left of the Observation viewer (under "Data"), click on the + next to the "level2" (or "level 2.5" if there is one)
- The names of the cube *context*s start with **HPS3D** (Herschel-PACS 3-dimension) and then contain the letters indicating the specific type of cube held in therein:
  - HPS3D**P**[R|B] are the red and blue ([R|B]) projected cube contexts
  - PS3D**R**[R|B] are the red and blue rebinned cube contexts
  - HPS3D**D**[R|B] are the red and blue drizzled cubes contexts
  - HPS3D**I**[R|B] are the red and blue interpolated cubes contexts

  Click on the + next to the HPS3DXX to see their individual, numbered, cubes:

Figure 1.1. Some of the cubes of Level 2: the Rebinned (HPS3DR[B|R]), with one cube per wavelength range (the outside 0 and 1) and per pointing (the inside 0 to 13); and the Projected (HPS3DP[B|R]), with one cube per wavelength range (0 and 1: the pointings have been combined into a single cube)

A summary of the contents of all levels of an *ObservationContext* is given in the PPE. To summarise:

- **HPS3DR[R|B]**. The *context* of **rebinned cubes** of class *PacsRebinnedCube*. There is one cube per wavelength range and per pointing specified in the observing proposal. These cubes have an irregular spatial grid and a non-equidistant wavelength grid. The spaxels are 9.4". For observations of point or slightly extended sources, the correctly-calibrated spectrum must be extracted from these rebinned cubes.

- **HPS3DP[R|B]**. The *context* of **projected cubes** of class *SpectralSimpleCube* are mosaic cubes created for all mapping observations. There is one cube per wavelength range requested in the observing proposal. These cubes have a regular spatial grid but also a non-equidistant wavelength grid. The spaxels are 0.5" for pointed observations, and up to 3" for mapping observations (3" if there is no drizzled cube provided, otherwise the same size as the spaxels of the related drizzled cube).

- **HPS3DD[R|B]**. The *context* of **drizzled cubes** of class *SpectralSimpleCube* and are mosaic cubes created for lineScan Nyquist and oversampled mapping observations, i.e. those with:

  - *Nyquist mapping:* in the blue, step sizes of up to 16" with a 3x3 raster; in the red step sizes of up to 24" with a 2x2 raster

  - *Oversampled mapping:* in the blue, step sizes of up to 3.0" with a 3x3 raster; in the red 4.5" step sizes with a 2x2 raster

  There is one cube per wavelength range specified in the observing proposal. These cubes have a regular spatial grid but also a non-equidistant wavelength grid. These cubes have a spaxel size that depends on the spatial sampling of the raster and the wavelength.

  Warning: you should not use the drizzled cubes from SPG 13 as the fluxes are incorrect. Use the projected cubes instead. Those of SPG 14 have correct fluxes.

- **HPS3DI[R|B]**. The **interpolated cubes** of class *SpectralSimpleCube* and are mosaic cubes created for pointed and tiling observations. There is one cube per wavelength range specified in the observing proposal. These cubes have a regular spatial grid but also a non-equidistant wavelength grid. They always have a spaxel size of 4.7" (SPG 13) or 3" (SPG 14). Undersampled mapping is

any mapping mode for which the steps sizes are larger than the Nyquist values given above, or the number of steps less, for each camera independently.

- If there is a Level 2.5 in the observation, the same cubes discussed here are provided but with a "BS" (background subtracted) appended to the end of the name

Go to the + next to the HPS3DXX. The list of numbers (+0, +1, +2...) are the individual cubes. A tooltip appears when you hover over a cube detailing its contents (wavelength range, position in raster, type of cube). Click on the numbers to see the associated cube open, either within the Observation Viewer, or to see it in a new window, right-click on the number and chose the **Spectrum Explorer** (SE: see the *DAG* chap. 6). Using the SE you can look at the spectra of your spaxels, perform mathematical operations, extract spectra or sub-cubes, make velocity and flux maps and fit your spectra. You can also drag-and-drop the cubes (to the *Variables* panel) to take them out of the *ObservationContext* and from there export as FITS.

See Section 10.6 to learn more about the native footprint of the PACS integral field unit, and that entire chapter to learn about the various cubes provided. (To find the observing mode of your observation, see Section 1.3.)

Note that if the observer did not request an off-source observation, then the final science products for unchopped rangeScans will be in the Level 2 of the observation, rather than Level 2.5.

1.4.3. The spectrum tables

From Track 13 we provide a table of the data of the rebinned cubes, which can be found in the *context*s called **HPSTBR[R|B]**. Within each *context* there is one table for each requested wavelength range of the observation, and all the spaxels *and all pointings* (for mapping observations) are contained in each table. The columns of data include the spaxel coordinates, the raster coordinate, as well as the wavelengths, fluxes and stddev values. See the PPE for a more detailed explanation of this table.

From Track 14 we provide a table of extracted spectra: **HPSSPEC[R|B]** at **Level 2/2.5** and **HPSSPEC** at **Level 3**. The first you can find for all pointed observations, and the second only for the subset of pointed observations that are chopNod and were taken to cover the entire SED in two or three separate observations. For this second, the data from both cameras and all the SED obsids are included in a single table (i.e. there is no [R|B]). The data in these tables are: the spectrum taken directly from the central spaxel, the point source-calibrated spectra "c1", "c9" and "c129" from the task extractCentralSpectrum for the chopNod AOTS and "c1" and "c9" for the unchopped AOTs. The spectra are converted into columns in the tables, and column names and Meta data help you understand the tables. See the PPE for a more detailed explanation.

1.4.4. The standalone browse products

The standalone browse products are created from the Level 2/2.5 cubes, and the Level 2/2.5/3 tables produced by the SPG, all of which are explained above. The standalone products are: the interpolated, drizzled, and/or projected cubes (depending on the AOT) but with an equidistant wavelength grid (i.e. each bin in the spectral grid is the same size); the data of the rebinned cubes as a table. All are provided as FITS files.

The reason for providing cubes with an equidistant wavelength grid is that these cubes then have a full WCS, with equidistant grids in the two spatial *and* the spectral direction. Cube viewers (e.g. ds9) can deal nicely with these cubes. A table of the data of the rebinned cubes is also provided: these cubes can never have a regular spatial grid, but by providing the spectral data as a table, the user can read them into most other software and deal with them perhaps more easily than in cube format.

These standalone products can either be downloaded directly from the HSA, or can be extracted from an *ObservationContext*: in the **browseProduct** level or in the Level they come from. They have the same name as the cubes discussed above but with an "EQ" added:

- **HPS3DEQP[R|B]**. The **projected cube** *context* in with an equidistant wavelength grid

- **HPS3DEQD[R|B]**. The **drizzled cube** *context* with an equidistant wavelength grid
- **HPS3DEQI[R|B]**. The **interpolated cube** *context* with an equidistant wavelength grid
- **HPSTBR[R|B]**. The **rebinned cube** *contexts*, with the data in a tabular format rather than a 3d product
- **HPSSPEC[R|B]**. The extracted spectrum table *contexts*, taken from Level 2 or 2.5, for all pointed observations
- **HPSSPEC**. The extracted spectrum table *context*, taken from Level 3, for all pointed, chopNod, full SED observations: concatenated table for red and blue camera and all obsids that contributed to the full SED coverage

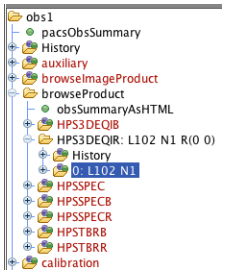

Figure 1.2. The standalone browse products (an example from a pointed observation from SPG 14.2)

Read the PPE for more information on these products.

1.5. I just want to look at my cubes!

Which Level 2 cubes should you look at?

- If you have a Level 2.5 in your ObservationContext, then you are looking at the background-subtracted on-source observation of an unchopped rangeScan observation-set and this Level contains the final cubes. Otherwise go to Level 2
- If you have a pointed observation of a point or semi-extended source, look at the rebinned or interpolated cubes (HPS3DR[R|B], HPS3DI[R|B]): the second are easier to load into software outside of HIPE because they have a regular spatial grid (which has been spatially-resampled from the first by the specInterpolate task), while the first have the native footprint of the PACS IFU
- Tiling observations: the projected or interpolated cubes (HPS3DP[R|B], HPS3DI[R|B]), where the second are generally preferred
- Mapping observations: drizzled or projected cubes (in that preference order) (HPS3DD[R|B], HPS3DP[R|B])

Extract the cubes from their contexts (the HPSXXX) with a drag-and-drop, or on the command line:

```
# To get the first level 2 blue drizzled cube
cubes = obs.refs["level2"].product.refs["HPS3DDB"].product.refs[0].product
# To get the second level 2.5 red projected cube
cubes = obs.refs["level2_5"].product.refs["HPS3DPBSR"].product.refs[1].product
```

1.5.1. Quick-look cube tools

There are a number of GUIs that can be used to inspect PACS cubes. These are explained in the DAG chps 6 and 7. When you have a cube highlighted in the *Variables* pane of HIPE (or in the directory listing in the Data panel of the Observation viewer) you can call up these tasks via the right-click menu. Note that all of these GUIs work on the individual cubes, not on the *context* they are contained

within (see Section 1.4), so you need to go down past the HPS3DXX level in the Level 2 layer of the *ObservationContext*, to the +0, +1...

- To scroll through 2D wavelength slices of your cubes you can use the **Standard Cube Viewer**.
- The **SpectrumExplorer** (see the *DAG* chap. 6). This is a spectral#spatial visualisation tool for spectra and cubes. It allows for an inspection and comparison of spectra from individual spaxels or from separate cubes, and it gives access to various mathematical tasks via its **Spectral Toolbox** menu (see chap. 6.4).
- The **Cube ToolBox**, which you also access via the SpectrumExplorer: see the *DAG* (chap 6.4, chap 6.5). Together with the Spectrum Toolbox, this allows you to inspect a cube spatially and spectrally at the same time. It also has analyses tasks#you can make line flux maps, velocity maps, and extract out spectral and spatial regions.
- The **Spectrum Fitter GUI**, which you also access via the SpectrumExplorer: see the *DAG* chap. 7. This GUI allows you to fit the spectra of your cubes with a variety of models. (For cubes it should be accessed via the Toolbox menu of the Spectrum Explorer, **not** directly from the HIPE *Tasks* panel.)

1.5.2. Create quick-look images or spectra from the cubes

A few quick inspection tasks so you can get a feel for your cube data:

- **Extract the spectrum of a single spaxel from a pointed observation/rebinned cube** with the task extractSpaxelSpectrum, which will only work a rebinned cube (HSP3D**R**[R|B]).

```
slicedRebinnedCube = obs.refs["level25"].product.refs["HPS3DRR"].product
spaxelX, spaxelY = 2,2
slice = 0
spectrum = extractSpaxelSpectrum(slicedFinalCubes, slice=slice,\
  spaxelX=spaxelX, spaxelY=spaxelY)
```

  To know which slice you want (i.e which cube in the *context*), you can use the task slicedSummary(slicedRebinnedCube). To know which spaxel you want to extract, open the cube with the Standard Cube Viewer, and the spaxelX and Y coordinates (in that order) of the spaxel under the mouse can be found at the bottom-left of the viewer. The "spectrum" output can be opened in the Spectrum Explorer.

- **Extract the spectrum of a single spaxel from a mapping observation/any cube** with the cube GUIs provided in HIPE. Extract the cube, e.g. for the second cube

```
cube = obs.refs["level2"].product.refs["HPS3DPR"].product.refs[1].product
```

  Open the cube in the **Spectrum Explorer** (right-click menu on "cube" in the Variables panel of HIPE), and from there select the **Cube Toolbox** (the  icon at the top of the "SE"), which will open in the top part of the SE.

  The Cube Toolbox tasks are located in the drop-down menu to the right of the plot panel: to extract a single spectrum use "extractRegionSpectrum", where you can select out a single spaxel with a click on the cube image. See the *DAG* (chap 6.7.3) to learn more about using the SE and the Cube Toolbox.

- **Extract the summed or average spectrum of a region**: using the Cube Toolbox mentioned above, you can also select a rectangular or circular region using the task extractRegionSpectrum: see the *DAG* (chap 6.7.5). This task will work on the Level 2/2.5 cubes: HSP3D[R|D|I|P][R|B].
- **Extract an image** of a single wavelength point is done the most rapidly with the Standard Cube Viewer, on any mosaic cube, right-click menu "Extract current layer".

- **Extract a wavelength-integrated image**: following the steps for "Extract the spectrum of a single spaxel" above to select the correct cube and open the Cube Toolbox, from there select the task "IntegrateSpectralMap". See *DAG* (chap 6.7.10.1) to learn how this works.

- **Plot the spectrum of a spaxel together with the RMS estimate**: using the pipeline helper task "plotCubesStddev" on any rebinned cube (HPS3DR[R|B]). See Section 7.7.1 for a longer explanation of this task.

1.6. Useful scripts and cookbooks for working with PACS data

There are useful scripts provided for working with PACS spectroscopy that can be obtained via the HIPE Scripts menu. Those of most interest to the non-pipeline processing astronomer are:

- **Fitting PACS cubes**: three scripts are provided for fitting a spectral line in PACS cubes and making images from the fitting results, e.g. integrated flux and velocity. These scripts are:

  1. *Spectroscopy: Fitting mapping observations (mosaic cubes)*. For the mosaic cubes of mapping observations, and starting from the interpolated, drizzled, or projected cubes (HPS3D[**I,D,P**][R|B]).

  2. *Spectroscopy: Fitting mapping observations (pre-mosaic cubes)*. Also for mapping observations but this time starting from the rebinned cubes (HPS3D**R**[R|B]). The difference with the previous script is that the fitting is done on these individual cubes of the raster and then the mosaicking is done on the fitting result images, i.e. creating 2d mosaics (images) rather than 3d mosaics (cubes).

  3. *Spectroscopy: Fitting single pointing cubes*. For pointed observations, creating fitting imges is more qualitative than quantitative, but nonetheless is useful for visualising the results for extended sources observed as a single pointing. The script starts with the interpolated cubes (HPS3D**I**[R|B]).

- **Point sources**: for point sources it is necessary to use the tasks provided to produce a correctly-calibrated spectrum of the point source from the rebinned cubes (those in HPS3DR[R|B] in the observation). The two scripts are: *Spectroscopy: Point source loss correction (central spaxel)* and *Spectroscopy: Point source loss correction (any spaxel)*.

- *Spectroscopy: Combine PACS and SPIRE spectra*: is a script that is aimed at observations of point sources, where you wish to combine the spectrum of these two instruments into a single spectrum. Note: this is not a mathematical combination, the spectra are simply stored in a single product, for ease of viewing and transporting.

- *Spectroscopy: Convolution for spectral images*: this script shows you how to take two spectral images (e.g. as created in the fitting scripts) and convolve the shorter wavelength image to the beam of the longer wavelength image: the images can then be directly compared to each other.

We are in the process of writing Cookbooks to explain some of the more difficult aspects of working with PACS spectroscopy, and to take the reader through some of the more common work-flows. These will be provided on the PACS "documentation" page on the HSC web-site: herschel.esac.esa.int/twiki/bin/view/Public/PacsCalibrationWeb.

Chapter 2. PACS Spectroscopy Launch Pad II

2.1. Introduction

This chapter should be read by those wondering whether it is necessary to run the pipeline on their data themselves. *For many observations, the SPG will produce good results and reprocessing will produce little change. However, note that the pipeline processing results do depend to a degree on the observations themselves (on the "observing conditions", if you like, and the brightness and complexity of the source)*. Hence, a re-processing for the more complex sources may improve the results. A detailed inspection of the spectra of the *PacsRebinnedCube*s (HPS3DR[R|B]) of Level 2/2.5 is strongly recommended (for *all* AOTs), before deciding whether to reprocess or not.

This chapter should answer the following questions:

1. Where and what are the PACS spectroscopy pipelines?
2. For what observations do I *need* to re-pipeline the data?
3. For what observations will it be *useful* to re-process the data, to try to achieve a better result?
4. Which are the *crucial pipeline tasks that have the greatest effect* on the resulting spectra?
5. I have a point source: what do I do next?
6. I have an extended source: what do I do next?
7. Where do I go to learn about the errors in my spectra?

2.2. The PACS pipelines

2.2.1. The SPG scripts: the automatic pipeline run by the HSC

The SPG scripts are those run at the HSC in automatic mode. The SPG treats the following AOTs separately: unchopped line and wavelength switching, unchopped range, chopNod line, chopNod range. The SPG processing takes a few months to run and hence there is a lag between a release of HIPE and the availability of those SPG products in the HSA. The very final processing of PACS data is that run in HIPE 14.2, for which this PDRG is written.

To know which SPG your downloaded observation was processed with, look at the Summary tab for the observation in the Observation viewer, or, for an observation called "obs",

```
print obs.meta["creator"]
```

The SPG scripts used within any track of HIPE are available from the HIPE Pipeline menu, within each of the submenus described below. They are not intended for interactive use#the interactive scripts should be used#rather they are provided for completness. In HIPE 14 the SPG scripts are based on the **Telescope normalisation** pipeline for the chopNod observations, and the **Calibration source and RSRF** pipeline for the unchopped observations. There are no differences in the range of tasks run in the SPG pipeline and the equivalent interactive ones.

2.2.2. The interactive pipeline scripts

The pipeline menu in HIPE is split into five: chopNod line, chopNod range, unchopped line, unchopped range, wavelength switching (an old mode, which uses the unchopped line pipeline script and which we do not discuss separately further). Inside each of these are a choice of pipeline scripts.

For the **chopNod modes**:

1. Using the telescope background spectrum for the flux calibration (**Telescope Normalisation**; this is the default script), and that which the SPG script is based on

2. Producing drizzled cubes for lineScan mapping observations (**Telescope normalisation drizzled maps**), and which is also incorporated in the SPG script

3. Using the calibration block and RSRF to flux calibrate the data, including producing drizzled cubes (**Calibration source and RSRF**)

4. Pointing offset corrections for bright point sources, reducing with the telescope normalisation method (**Pointing offset correction (point sources)**)

5. Comparing on-source spectra to off-source spectra; a "helper" script, not a full pipeline (**Split On-Off**)

6. **Combine observations for a full SED**, also a "helper" script, runs the pipeline on multiple obsids and combines the results at the end (rangeScan only)

For the **unchopped modes**:

1. Using the calibration block to flux calibrate the data (**Calibration source and RSRF**). Includes a "transient correction" task for the lineScan AOTs only, and is the script on which the SPG script is based

2. Using the calibration block to flux calibrate the data and apply a more effective transient correction, for line and rangeScan (**...with transient correction**)

3. Pipeline process the on-source and off-source observations, and then subtract them (**Combing off-source with on-source**: rangeScan only). There is also a PACS useful script offered via the HIPE *Scripts* menu do to this subtraction (*Scripts#PACS Useful scripts#Spectroscopy: Off-subtraction and post-processing in unchopped range spectroscopy*)

How many wavelength ranges are included in your observation, and whether you have a single pointing or a mapping observation does not matter: all pipelines handle all these cases.

To learn more about these pipeline scripts and their differences: see Section 3.4.2.

To access the scripts: go to the HIPE menu *Pipelines#PACS#Spectrometer*. The scripts assume

- You know the "obsid" of your observation
- You have the calibration files on disk; normally you will use the latest update (updates are searched for automatically when you start HIPE)
- You do the red and the blue camera separately

To run the scripts,

- Read the instructions at the top, and at least skim-read the entire script before running it
- It is *highly* recommended you run line by line (at least the first time)
- To be able to edit and save the script, save the script to a new, personalised location: otherwise you are changing the script that comes with your HIPE installation. *However*, as these scripts evolve

with time, do not blindly continue to use that pipeline script for future processing: always check against the latest release of HIPE

As you run the scripts,

- Plotting and printing tasks are included, with which you can inspect the data layout or the spectra themselves
- You will be offered various ways to save the intermediate data products to a pool on disk, but saving cubes or spectra as FITS files is only possible towards the end of the pipeline, when single FITS-able products are created

In addition, the SPG scripts are provided in each of the pipeline sub-menus.

2.3. For what observations *must* I re-pipeline the data?

There are only a few cases where it is definitely necessary to re-pipeline the data.

1. **Spectral lines at wavelengths redder than 190μm,** it will be necessary to run the "calibration sources and RSRF" scripts for these observations, and moreover using a very particular version of the RSRF (relative spectral response function). See Section 5.4 (chopNod) or Section 6.4 (unchopped) for more explanation. You can re-reduce your data from Level 0.5 (if working from an SPG >=13 observation).

2. **Updates to the calibrations** (especially if your previous reduction was several HIPE version ago). Any changes to the calibration between the time the data you have were processed and the current status will require running some or all of the pipeline. You can consult the "What's New" pages to find out what is new in each track (e.g. herschel.esac.esa.int/twiki/bin/view/Public/HipeWhatsNew14x for Track 14), and to learn how to install calibration updates go to Section 3.6.

Note: in SPG 13 and earlier, the rangeScan observations were not flatfielded by the SPG. In SPG 14 this is now done: ranges of less than about 5 microns are flatfielded with the lineScan task and those of longer with a new version of the rangeScan flatfielding task. See the pipeline chapters to learn more.

2.4. For what observations is it *useful* to re-process the data?

For certain types of observations it is useful to try out a different pipeline script to that run by the SPG: to see if you can get a better result, to check for contamination; or to try to improve the results of some of the crucial pipeline tasks.

1. **Bright point sources**: with continuum flux levels of order 10s Jy and more. A special pipeline script for these brighter point sources where the source is located with the central 3x3 spaxels—and ideally close to the central one—is provided for pointed chopNod mode observations. This end result of this script is a calibrated spectrum of the point source, and this spectrum should be cleaner than the standard, point-source calibrated spectrum that you can obtain yourself from the cubes created by the SPG or from the spectrum tables provided in the *ObservationContext*. The script is called **Pointing offset correction (point sources)**. You can start the pipeline from Level 0.5 if working from an SPG >=13 observation. Do compared the results of this script to that of the standard script. The pipeline is explained in Section 5.2.

2. **Check for contamination in the off-source pointings** for chopNod mode observations using the script **Split on-off** scripts. If the continuum level in the off-cubes is higher than in the on cubes, or spectral lines are visible, then you probably have contamination.

Another way to check for off-source contamination for chopNod mode AOTs is to run the **Telescope Normalisation** and the **Calibration source and RSRF** pipeline scripts and compare their results (using the rebinned cubes for the comparison). If there is line emission in the off-source spectra, then the spectral lines will look very different in the two sets of resulting cubes. In this case, you should favour the result from the calibration block pipeline, *while noting that this will not "get rid" of the contamination, it will at least not exaggerate its effect*.

For unchopped rangeScan observations, checking the off-source data is straightforward: compare the Level 2 rebinned cubes (HPS3DR[R|B]) from the off-source observation to the same cubes for the on-source observation. For unchopped lineScans, doing this comparison is also simple but requires a few additional lines of code as the appropriate products are not immediately available in the *ObservationContext*, but need to created. This is explained in Section 10.9.

3. **Transients correction for unchopped AOTs**: unchopped rangeScan AOTs have no transient correction in the SPG scripts, and that in the lineScan script is a first version of the correction method. A better transient correction can be found in the **...with transient correction** pipeline scripts, for line and rangeScan observations. Transients are short or mid-term effects that change the response of the detector (e.g. following cosmic ray hits), and so affect the flux levels of the spectra over short and intermediate time-scales. *Please note that new transient correction tasks are interactive and will require more than a blind running of the script*.

   While there is no transient correction in the SPG script for unchopped range scans and only a basic one for line scans, it turns out that the flatfielding (which *is* done in both scripts) does a good job of fixing transients anyway.

4. **Complex spectra**: with many emission lines, with absorption lines, faint spectra, or those with broad-band features. It is worth checking the results of the flatfielding task for these spectra (for this you need to re-run the pipeline: see below), possibly adjusting the flatfielding to deal better with the presence of these features.

2.5. Which are the crucial pipeline tasks?

1. **Spectral flatfielding**. The flatfielding can be a crucial task for improving the SNR of the final spectra. The flatfielding operates in the spectral domain. For each spaxel of a cube there are several discrete spectra that need to be averaged to create the final single spectrum. If some of the discrete spectra are discrepant in signal level, the flatfielding will correct this, and so improve the SNR of the subsequently-averaged spectrum, and should also smooth the continuum shape.

   For very faint targets (continuum of only a few Jy) it is worth checking the results by performing the flatfielding yourself, with the "verbose" pipeline script parameter to set True. For faint spectra, the correction is very difficult to compute when the continuum is near 0, and hence it is also worth checking that the results are reasonable—if the continuum of the flatfielded spectra have the same SNR as those before, it is even not worth doing a flatfielding at all.

   For targets with more than one line in the spectrum, or those with absorption lines it is worth checking the results by performing the flatfielding yourself, with the "verbose" pipeline script parameter to set True. The flatfielding masks out lines as part of its process, but for spectra with multiple lines or absorption lines it may help to specify the lines to mask out via an input line list.

   Flatfielding for rangeScans the longer wavelength range of the SEDs can be difficult to flatfield well if there are many "curves" in the spectrum, or strong slopes in the fluxes. It could be worth re-running the flatfielding for such observations. However, in HIPE 14.2 the default flatfielding fitting function is a spline, compared to a polynomial previously, and this deals with curvature quite well.

   To run the flatfielding, you need to run the pipeline script from Level 0.5. The flatfielding steps in the pipeline scripts are explained in Section 5.2.6 and Section 5.2.7 (chopNod line, range), Section 6.2.7 and Section 6.2.8 (unchopped line, range). How to compare different flatfielding attempts, and some more things to pay attention to is explained in Section 7.4.

2. **Wavelength grid**. The wavelength grid used in the SPG pipeline (which is also the default in the pipeline scripts) has been chosen to give the best-looking spectra for most observations. The wavelength grid of the final cubes is created from the individual grids that are present in each of the 16 pixels that feed each spaxel, each of which is slightly offset from the others. The pipeline regularises this collection of grids into a single one, common to all spaxels and all cubes of the same wavelength setting. This regularised grid is created by the task "wavelengthGrid" with the aid of two parameters—`oversample` and `upsample`—which determine the final spectral sampling and which data-points are sampled for each bin. *It is important to note that if upsample is #1 (which it is in the SPG), the signal in the bins then become correlated as some of the same data-points feed neighbouring bins*. If this is a problem you will need to re-run the pipeline from Level 0.5 until the end, and chose the parameters of the wavelength grid differently. The interplay and effect of choosing different values of `oversample` and `upsample` is explained in more detail in Section 7.2.

3. **For the unchopped modes (line and range), for those with more than one off-source pointing, the background subtraction is crucial**. If the background level changes—a not uncommon occurrence because of the effect of transients—then the background subtraction will suffer. The task that does this, "specSubtractOffPosition", offers three algorithms for computing the background to subtract, and it is recommended you experiment with these. For this you need to re-run the pipeline from Level 1: the unchopped pipelines are explained in Chapter 6, and some examples of background subtraction are included in Section 7.5.

2.6. Point and slightly-extended sources

For point sources, stop the pipeline at the creation of the final rebinned cubes (called "slicedFinalCubes" or "slicedDiffCubes", or HPS3DR[BS][R|B] in Level 2/2.5 of the *ObservationContext*). Then you *have to* extract your point source spectrum using one of the tasks provided to do this, since you *have to* apply the point source flux correction. These tasks are explained in Section 8.4 (not-centred point sources) and Section 8.5 (centred point sources) and their use in the pipeline can be found in Section 4.3.

The post-processing tasks for point sources are part of the pipeline scripts, and hence are documented using the terminology and product names of the pipeline (Section 4.3). However, if you want to try some of these tasks on your SPG products, as just gotten from the HSA, we have created two useful scripts that show how to run these tasks: on a pipeline product or on any observation gotten from the HSA: *Scripts#PACS useful scripts#Spectroscopy: Point source loss correction (any spaxel)* and *Spectroscopy: Point source loss correction (central spaxel)*.

A few tips:

1. *For bright point sources* with continuum levels exceeding 10 Jy, we also offer the **Pointing offset correction (point sources)** interactive script for chopNod observations. The resulting spectrum should be cleaner than that gotten from the SPG. The output of this script is the "c9" or "c129" from extractCentralSpectrum. This task is explained in Section 4.3.

2. *For very faint point sources*, where there is more noise than signal in the spaxels surrounding the central one, use the output "c1" from the task extractCentralSpectrum (Section 8.5). Otherwise use "c129".

3. *For point sources with an uncertain continuum level*: any unchopped mode observation or for spectra longwards of 190 μm: use only the result "c1" or "c9" produced by extractCentralSpectrum (see Section 8.5).

4. *For point sources where you have SPIRE data, or several PACS bands*, and you want to push the extracted point source spectra together into a single product, you can use a useful script *Scripts#PACS useful scripts#Spectroscopy: combine PACS and SPIRE spectra*. This script is explained in Section 8.2.

For semi-extended sources, i.e. those that are still entirely contained within the central 3x3 spaxels of the IFU (radius less than around 20 arcsec) and for which you have a good idea of the source

morphology, and which is centred on the central spaxel, there is a task to extract and calibrate its spectrum. See Section 8.6.

2.7. Extended sources

For extended sources there is a choice of several different types of cubes that can be created. This depends on the AOT of the observation: whether pointed or mapping, what type of mapping observation it was, and whether it is a lineScan or a rangeScan. See Section 1.3 and Section 3.3 for advice on how to know the AOT. For mapping observations, the end result of the pipeline is usually a "mosaic" cube, of which we offer three: *drizzled, projected, and interpolated*. For pointed observations, the end result is either the same type of cube as used for point source work (a *rebinned* cube) or a spatially-resampled version of that (*interpolated* cube).

If you want to create a cube that is not present in the *ObservationContext* gotten from the HSA (since only the two best of the three possible cubes are created), or re-create cubes with a different spaxel size, but do not wish to run a full pipeline script, you can follow two PACS Useful scripts:*Scripts#PACS useful scripts#Spectroscopy: Post-processing for extended sources*, and *Spectroscopy: Re-create the standalone browse products*.

A few tips:

1. Drizzle is optimised for spectral regions of 1 or 2 microns, and it may take a long time to run. **Drizzle results for HIPE 13, from the HSA or those created using the pipeline scripts had incorrect fluxes. For HIPE 14 this has been corrected**.

2. SpecInterpolate is not recommended for oversampled maps, and while it is not optimised for Nyquist sampled maps, the results can be used (compare to the specProject results first).

3. For single pointings, bear in mind always that the beam is *very much* undersampled: the spaxels are not small enough to fit at least 2 or 3 of them within the FWHM of the beam. This means that there are gaps in the spatial coverage, which will result in a loss of information—flux—such that the true source morphology can never be fully reconstructed and flux not recovered. How important this is depends on the size and morphology of your source. It is recommended that you compare your results to those obtained from photometry where-ever possible.

Once you know what cube you want to work with, and where to get it from—Level 2.5 for unchopped range observations, Level 2 for all others—you can then work with your cubes with various tools in HIPE. Some useful scripts and cookbooks have been written to introduce you to the tools that HIPE provides for inspecting and fitting cubes: see Section 1.6.

2.8. Where can I learn about the errors in my spectra?

An explanation of how error estimates are provided for PACS spectroscopy is given in Section 7.7. You are encouraged to read this section carefully.

All the cubes at Level 2/2.5 have an error array, but they are created differently.

- Rebinned cubes (HPS3DR[R|B]), aka *PacsRebinnedCube*s. These cubes are created from *PacsCube*s. In each spaxel of a *PacsCube* there is a collection of (at least 16x2) spectra, all of the same wavelength range but gathered by 16 different instrument detectors and during subsequent runs on the grating. These individual spectra are combined into a single spectrum, for each spaxel, by averaging along an input regular wavelength grid, with bin-sizes slightly larger than the separation of the data-points and optimised to the wavelength. Each bin in the wavelength grid of the output rebinned cube, therefore, has a contribution from several data-points from the input *PacsCube*. The scatter in these data-points is computed and becomes the "stddev" array of the rebinned cube. Since PACS spectral sampling is highly redundant over most of the spectral range, this can be considered

a measure of the noise in the input spectrum due to a combination of stochastic (photon) noise and instrumental noise.

The task that creates the wavelength grid that is used for the rebinned cube has the parameters "upsample" and "oversample" that determine (i) the bin sizes as a fraction of the spectral resolution and (ii) the stepping forward along the input *PacsCube* wavelength array as the data-values of each bin are computed. These two parameters affect the spectral sampling and also the degree of dependence of the bins on their neighbours (i.e. how many data-points are shared between bins). Different values of upsample and oversample will result in a slightly different appearance of the resulting spectra—the line shapes and the apparent noise. The values chosen by the SPG pipeline are determined by the density (degree of redundancy) of the spectra (which was set by the observer). This is explained in more detail in Section 7.2.

- Drizzled cubes (HPS3DD[R|B]) are created by the task drizzle, which works on *PacsCubes*. It also takes the scatter in the *PacsCubes* data-points along the same wavelength grid used to create the rebinned cubes, and propagates them using the standard error propagation rules, slightly modified to work with the drizzle method. For more information on how drizzle handles errors, see Section 7.8. The result is placed in an array called "error".

- Projected cubes (HPS3DP[R|B]) are created by the task specProject. This task propagates the stddev array of the rebinned cubes using standard error propagation, i.e. for each projected spaxel (which are smaller than the spaxels of the rebinned cubes), the error is SQRT([(stddev of rebinned spaxel) x (weight of the projection of that rebinned spaxel onto the projected output pixel)]^2). The result is placed in an array called "error".

- Interpolated cubes (HPS3DI[R|B]) have an error array created by an interpolation of the stddev array of the rebinned cube, using the same algorithm that interpolates the flux array. The interpolation error (specInterpolate *estimates* the fluxes between irregularly gridded datapoints) is not included.

For point sources extracted from rebinned cubes using the point-source tasks, you will find the following:

- extractCentralSpectrum: works on the rebinned cubes. The stddev array(s) of these cubes is(are) propagated by the standard rules, e.g. the errors are combined for the output spectra which are a combination of several spaxels (c9 and c129). The output spectra all have a weights array, which is the propagated stddev of the rebinned cube but inverted: 1/stddev**2

- extractSpaxelSpectrum: works on the rebinned cubes, but does not propagate the stddev; the output has no weights or error array. The associated tasks (undoExtendedSourceCorrection, pointSourceLossCorrection) used when the extracted spaxel spectrum is to be point-source calibrated, therefore neither propagate errors.

Calibration uncertainties are *not* included in these errors.